IDnow GmbH
Identity.TM Web Services

API Reference
Version 2.10af

Index

1 Changelog

Version	Date	Editor	Changes
2.0	2017-03-06	Markus Herzog	Initial new Version
	2017-03-16	Markus Herzog	New function “getVoiceFiles”
2.01	2017-04-26	Markus Herzog	New function “getVideoQueueStatus” Added Status 89 to ExtendedStatus List
2.02	2017-05-17	Markus Herzog	Added object RedirectURL on putOrder Changed putOrder.Add possible Options
2.03	2017-05-30	Markus Herzog	New function “getShops”
2.031	2017-07-31	Markus Herzog	New function “getVideoFileBinary” Added object WebhookURL on putOrder
2.032	2017-08-18	Markus Herzog	New function “getESignHash”
2.032b	2017-11-20	Markus Herzog	Added new type for Additional Field Country ISO on putOrder
2.032c	2018-02-12	Markus Herzog	Error in definition getIdentData->IDCard->Type corrected
2.032d	2018-03-20	Markus Herzog	Added object eSignHash on putOrder
2.04	2018-05-28	Markus Herzog	New function “checkSignmeUser”
2.05		Markus Herzog	Added product autoID
2.06	2018-10-04	Erkan Ciftci	Removed no longer supported endpoints removed deprecated putOrder elements added several descriptions and sign-me features
2.06b	2018-11-29	Erkan Ciftci	Minor corrections and clarifications
2.07	2018-10-09	Markus Herzog	new endpoint added error codes on some errors extended
2.08	2019-01-04	Markus Herzog	Internal Processing changes
2.09	2019-01-21	Markus Herzog	Added element AllowedProductSet to putOrder Added field DateOfIssue to IdentData->IDCard
2.09a	2019-03-01	Erkan Ciftci	Approved Version
2.09b	2020-05-20	Timo Tegtmeier	Minor clarifications and extensions
2.09c	2020-06-20	Timo Tegtmeier	Added information about e-mail notifications
2.10	2020-08-23	Timo Tegtmeier	Editorial overhaul, added new PoS functionality
2.10a	2020-08-24	Timo Tegtmeier	Added eID steering and mobile phone verification
2.10b	2020-09-08	Timo Tegtmeier	Added eidRestrictedIdentifier to getIdentData result
2.10c	2020-09-10	Timo Tegtmeier	Added information about WebHook retry scheme Added simplified flow diagrams for video identification and eSigning
2.10d	2020-09-21	Timo Tegtmeier	Added monitoring call “systemStatus” Added notes about multi signing to “putOrder” Minor formatting changes
2.10e	2020-10-21	Timo Tegtmeier	Added AES functionality to “putOrder” Added “Lang” attribute to “putOrder” Added information about previously undocumented webhooks Added “getDeepLink” attribute to “putOrder” Extended monitoring information Added field “extDocumentType” to “getIdentData” response Added crypted data retrieval
2.10f	2020-11-30	Timo Tegtmeier	Added “DisableEID” to “eIDSettings” in “putOrder”

			Added visible signature control options for multi-sign orders to “putOrder” Added “purpose” element in “putOrder” description Minor corrections
2.10g	2021-02-01	Timo Tegtmeier	Minor (textual) clarifications Corrected information about supported browsers Added “getAutoIdLink” to “putOrder” Added section about deep linking
2.10h	2021-02-12	Timo Tegtmeier	Added new fields “OcrStatus” and “eIdUsed” to “getIdentData” Added field “countryOfIssue” in “IDCard” to “putOrder” Added function “getESignAuditLogPDF“
2.10i	2021-02-23	Timo Tegtmeier	Added new product “Mobile Number Verification” Added configurable eSign TAN SMS Added field “countryOfIssue” in “IDCard” to “getIdentData” Added new web hook “OnStatusChange”
2.10j	2021-04-10	Timo Tegtmeier	Added sign-me feature “Basic Electronic Signature” Added new parameter “signatureType” to “checkSignmeUser” Documented pre-existing sign-me feature to create an account only if no account exists Document pre-existing methods to request a password change Added clarification regarding future versions of this API
2.10k	2021-05-19	Timo Tegtmeier	Added eSign stamping feature Added “SecurityCheckFailed” flag to “getIdentData” Extended “extDocumentType” definition to cover the new German eID-Card for EU citizens
2.10l	2021-07-01	Timo Tegtmeier	Editorial changes Updated and extended list of status codes Completely re-written chapter about status codes
2.10m	2021-08-06	Timo Tegtmeier	Added option to retrieve sub-status codes Added field “DateOfIssue”
2.10n	2021-10-19	Timo Tegtmeier	Minor editorial changes Added new function “getImages”
2.10o	-	-	Willingly skipped to avoid confusion (2.10o vs. 2.100)
2.10p	2022-01-07	Timo Tegtmeier	Changed company’s legal entity type Fixed bad references Minor editorial changes
2.10q	2022-03-28	Timo Tegtmeier	Renamed document Added shop self-service features Removed obsolete functions Extended systemStatus and added IDnow AutoIdent Corrected response data of “getIdentData” Added additional variant of “getIdentData”
2.10r	2022-09-09	Timo Tegtmeier	Updated Pusher URLs Sharpened possible response for field “IdentifyMethod” Added field “FraudAttempt” in “getIdentData” response
2.10s	2023-03-08	Markus Herzog	Rebranded to IDnow GmbH Removed shop features Added Product Shop Added Element CustomerRouteTag to putOrder Added new Gender value: “diverse” in IdentData Added “tag” feature for signature/stamping positions
2.10t	2023-05-23	Markus Herzog	Added eSignOptions->AdditionalCheckboxes to putOrder
2.10u	2023-07-07	Markus Herzog	Added new method getOrdersByRef
2.10v	2023-09-21	Markus Herzog	Added possibility to define signature/stamping Offset when using tag
2.10w	2023-11-07	Markus Herzog	Added scan 153 to getStatus/ExtendedList Extended field putOrder->CollectedBy to 255 chars
2.10x	2024-05-17	Markus Herzog	Added Product Reader

2.10y	2024-08-01	Markus Herzog	Remove Product Courier
2.10z	2025-01-03	Markus Herzog	Change for scan 103: from extended to default response (getStatus)
2.10aa	2025-04-15	Alexandros Chanizmpekov	Added Testrobot
2.10ab	2025-05-13	Alexandros Chanizmpekov	Added missing eID product with number 13 Added new option of IncludeEidJwt to getIdentData endpoint
2.10ac	2025-06-06	Alexandros Chanizmpekov	Added new root Autotest element to putOrder and updated the testrobot documentation accordingly.
2.10ad	2025-06-23	Markus Herzog	Added possibility to only request eidRestrictedIdentifier for eID Orders
2.10ae	2025-09-22	Avraam Kotanidis	Removed Product SelectIdent with number 0
2.10af	2026-01-21	Markus Herzog	Add SIGN flow to Testrobot

2 General Information

This document describes the usage of **identity.TM's by IDNow** customer-facing APIs. Intended audience are developers and system integrators.

The **identity.TM** Legal API and the deprecated eSignBackend API are covered in different documents.

The information owner is IDnow's CTO.

2.1 Contact and Credentials for Service Integration

Throughout the technical implementation you will be assisted by IDnow. itm-it-support@idnow.io is the contact for any kind of technical and procedural questions. If you have not already received backend credentials for the services in this document, please apply for some by contacting IDnow.

2.2 Future development of this API

This API is under constant development. New versions with new features can appear at any point in time without further notice.

Adding new elements to responses or optional elements to requests is considered to be a compatible change and neither the major nor minor version of the API will change. Only this document will be updated and carry a new index letter.

Because of this policy, you must not assume, that responses will carry only the elements outlined in a specific version.

In case of incompatible changes, the new API will receive either a new minor or major version number, depending on the size of the changes.

2.3 Basic definitions

User	Natural person to be identified, also referred to as “signee” in eSign context
OrderID	**identity.TM’s** internal unique reference to an order
UserID	**identity.TM’s** internal unique identification number for a shop agent
UserToken	**identity.TM’s** internal security token for agent sign-in purposes

Datatype Codes

O	Object
A	Array
DS	Date string, with format YYYY-MM-DD
DTS	Date time string according to ISO 8061 (e.g., 2013-07-15T18:00:00+02:00)
TS	Time string, 24-hour-based, format HH:MM
S	String
N	Number
B	Boolean
PN	International phone number without blanks (MSISDN) e.g., +491701234567
CC	The ISO-Code of the country (ISO 3166) e.g., DE, AT, NL

Abbreviations

Len	Length
Type	Datatype, abbreviated
Req	Element is required
Occ	Element occurs
Y	Yes
N	No
Dep	Depends on parameters/conditions

2.4 Typical Identification Flow

The identification of a natural person is started by creating an identification order. As a result, a URL will be returned. Typically, the user will now be redirected to that URL to choose between different identification methods (dubbed *Verfahrensseite* or landing page).

The user will then carry out the identification. In the meantime, you can lean back and wait for a result to come.

After the identification has been conducted, the user will typically be redirected to you (in the rare case that no redirect is desired, the user will be redirected to a friendly page on **identity.TM** side that thanks the user for conducting the identification).

Once a final judgment on the result of an identification has been made, you will be notified asynchronously via a webhook, that a result is now available. You would then make API calls to query the status of that specific identification order and query the identification data, if

available. Identification data will be available in case the identification has been conducted positively or the identification is suspicious to fraud.

Once all data has been retrieved, you will then make an API call to delete the identification data.

In case the user does not finish an identification, a configurable time out will occur, that invalidates and deletes the order. A webhook will be triggered to indicate that condition.

2.5 Redirects

By default, the user will be redirected to the designated URL when the identification is finished from an end-user perspective. Keep in mind, that this moment will most probably not be the moment in which the order is finally processed, so no result data might be available. You must wait for the `OnFinal` webhook to be triggered, before retrieving results.

Alternatively, your account can be configured to keep the user on a spinner page on **identity.TM** side until the order is completed. In this case, data will be available when the redirect happens. Please contact itsupport@identity.tm if you would like to have such a spinner or more information on this topic.

2.6 Webhooks

Like the redirect URLs, the webhook URLs can also be set up within each `putOrder` request (`WebhookURL` element) that is being made. If these URLs do not change for each order, it is possible to configure them on a per account level.

The placeholders `[Ref]` and/or `[OrderID]` can be used to make the URL dynamic. `[Ref]` will be replaced by your internal reference number as specified when creating the order while `[OrderID]` will be replaced with **identity.TM's** unique OrderID of the order.

For OnStatusChange Webhook URL you can also use the placeholder `[statusCode]`.

By default, Webhooks will be delivered via GET requests. This can be changed on a per account level to either POST or PUT if necessary.

2.6.1 Recommendations

Do not try to run longer operations in the code that receives the webhook. Webhooks will be executed synchronously, and further order processing will be blocked until a webhook call has been finished. The operation should not take longer than 5 seconds and the webhook call will be terminated after 10 seconds.

2.6.2 Retry scheme

In case a webhook is not replied with a HTTP status code 200 or 202, the system will try to deliver the webhook again. The first retry will be started after approximately 86,4 seconds after the first try.

From	To	Interval
86.4s	**864s**	Once every minute
864s	**12h**	Once every hour
12h	**6d**	Once at night

2.6.3 Example URLs

```
https://www.example.com/identity/callback/AfterVideo?orderID=[OrderID]&ref=[Ref]
https://www.example.com/identity/callback/OnFinal?orderID=[OrderID]&ref=[Ref]
```

2.7 Technical Data

Service Kind

RESTful web service over HTTPS (HTTP 1.1 status codes)
JSON Data in HTTP Body

Base URLs

Customer Services: https://customer.identity.tm/api/2.10/

Content-Type

application/json

Authentication

Customer Services: HTTP Basic Authentication

TLS

Version 1.2

3 Handling orders

3.1 Creating Orders

This central method is used to create any kind of new order for identifications and/or qualified electronic signatures for natural persons.

3.1.1 Special considerations for eSigning

3.1.1.1 Signature Type

You can choose between using AES (Advanced Electronic Signature) or QES (Qualified Electronic Signature) when creating an order to sign documents. When creating orders, QES will be used by default.

To be able to use AES for signing, you need to go through a special on-boarding process in which a special seal will be created that will later be used to create AES'. Please contact IDnow for that.

3.1.1.2 Multi-Signing

When requesting multiple parties to sign a document, please make sure, that each person uses a distinct mobile number. If the phone number is shared among two or more parties, the signing process will fail predictably.

Please also inform your customers about that restriction.

3.1.2 Requirements

The requested products must be associated with and enabled on your account by IDnow.

3.1.3 Method and URL

```
[PUT] https://customer.identity.tm/api/2.10/putOrder
```

3.1.4 JSON Request Body

Element	Parent	Type	Len	Req	Description
Ref	[ROOT]	S	100	N	Your internal reference number to the order. This field can be used to associate the order with an ID in your system and has no meaning for IDnow except in the case of a Testrobot (see 9.5).

Product	[ROOT]	N		Y	Specifies the product you want to use. **8** ShopIdent **12** Identification via any of the enabled methods including Video, but excluding autoID **13** eID Online-Ausweisfunktion (eID PIN, available for German documents only) **15** eSign **16** Identification via autoID **17** Mobile phone number verification **19** Reader
Add	[ROOT]	N		Y	Specifies additional product behaviors. 0 will be the correct choice in most cases. **0** Normal behavior **1** eID: only request eidRestrictedIdentifier **4** Sign-me: use existing account, if available, otherwise create new account **8** Sign-me: create account **16** Sign-me: use existing account **128** Use AES instead of QES for eSign **256** Use BES instead of QES for eSign (sign-me only) **4096** License Check (with ID) **8192** License Check (without ID) **131072** Record Video **4194304** PoS eSign service **8388608** PoS identification service
Lang	[ROOT]	S	2	N	Language hint for video system, **has no effect on web or app flow**. Use ISO 639-1 language codes.

eIDSettings	[ROOT]	O		N	Settings for eID, no effect if user selects different method
VerifyMobileNumber	eIDSettings	B		N	Enables mobile phone number verification
EnableOCR	eIDSettings	B		N	Enables OCR to gather document number, issuing authority and date of issue of identification document.
CaptureImages	eIDSettings	B		N	Capture images of the legitimation document; if EnableOCR is set, images will always be taken
NoVideoFallback	eIDSettings	B		N	If eID fails, no fallback to Video will be offered.
DisableEID	eIDSettings	B		N	Disable eID
getTargetLink	[ROOT]	B		N	Set to true to get the TargetURL in response (otherwise only the user will be informed via e-mail)
getVideoHashes	[ROOT]	B		N	Set to true to get the video system hash in the response if Product is 12 or 15.
getDeepLink	[ROOT]	B		N	If set to true, a deeplink for mobile application will be part of the response
RedirectURL	[ROOT]	O		N	
Success	RedirectURL	S	150	N	Redirect to URL after success
Abort	RedirectURL	S	150	N	Redirect to URL after failure
WebhookURL	[ROOT]	O		N	
AfterVideo	WebhookURL	S	250	N	Webhook to trigger when the user leaves the video system. Cannot be used together with AfterIdent. Ident data will not necessarily be ready when triggered.
AfterIdent	WebhookURL	S	250	N	Webhook to trigger when the identification has been conducted. Cannot be used together with AfterVideo. Ident data will not necessarily be ready when triggered.

OnFinal	WebhookURL	S	250	N	Webhook to trigger when the order has been finally processed
OnError	WebhookURL	S	250	N	Webhook to trigger when a partner is unable to process an order (does not apply to video)
OnVideoError	WebhookURL	S	250	N	Webhook to trigger when a temporary error occurs while performing a video identification
OnStatusChange	WebhookURL	S	250	N	Special webhook to trigger on when order reaches a specific status (configurable on account level)
Phone	[ROOT]	PN	50	N	Phone number of your customer
Mobile	[ROOT]	PN	50	N	Mobile number of your customer
Email	[ROOT]	S	50	Y	Email address of your customer
Address	[ROOT]	O		Y	Mandatory if user is allowed to use product Shop
Firstname	Address	S	50	Y	Physical Address data of your customer (Contact / Location)
Surname	Address	S	50	Y	
CompanyName	Address	S	50	N	
Street	Address	S	50	N	
ZipCode	Address	S	5	N	
City	Address	S	100	N	
Country	Address	S	3	Y	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL
IdentData	[ROOT]	O		N	The fields you provide, will be checked. Empty fields will be picked if possible.
Firstname	IdentData	S	50	Dep	Please give us notice if you plan to omit any of these fields.
Surname	IdentData	S	50	Dep	
Street	IdentData	S	100	Dep	
ZipCode	IdentData	S	5	Dep	
City	IdentData	S	100	Dep	

Country	IdentData	S	2	Dep	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL
Birthdate	IdentData	DS		Dep	Date of birth
Birthplace	IdentData	S	100	Dep	Place of birth
Birthname	IdentData	S	75	N	Surname at birth. Will only be determined if field is neither null nor omitted
Nationality	IdentData	S	50	Dep	Nationality
Title	IdentData	S	50	Dep	Title
Artistname	IdentData	S	50	Dep	Artist name/alias
Gender	IdentData	S		Dep	female/male/diverse
Email	IdentData	S	50	N	Email address of your customer
Mobile	IdentData	PN	50	N	Mobile number of your customer
CollectedBy	IdentData	S	255	Dep	Identifier who carried out the identification, PoS eSign Service only
IDCard	IdentData	O		No	Data about the ID document
CardNo	IDCard	S	50	Dep	ID document number
PlaceOfIssue	IDCard	S	100	Dep	ID document place of issue
DateOfIssue	IDCard	DS		Dep	ID document date of issue
CountryOfIssue	IDCard	S	3	Dep	The ISO-Code of the issuing country (ISO 3166) e.g., DE, AT, NL
ValidUntil	IDCard	DS		Dep	ID document valid until
Images	IDCard	O		Dep	
Front	Images	S		Dep	Base64 encoded JPEG image of the front side of ID Card
Back	Images	S		Dep	Base64 encoded JPEG image of the front side of ID Card
DriverLicence	IdentData	O		Dep	If Driver's License Check with Add 4096 and/or 8192, provide data if available
LicenceNo	DriverLicence	S	50	Dep	
Classes	DriverLicence	S	50	Dep	

CountryOfIssue	DriverLicence	S	3	Dep	The ISO-Code of the issuing country (ISO 3166) e.g., DE, AT, NL
DateOfIssue	DriverLicence	DS		Dep	Date of issue
ValidUntil	DriverLicence	DS		Dep	Date until the license is valid
Additional	IdentData	O		N	If you need special attributes, you can add here for Shop up to 2, Video up to 10 items. <detail> can have a length of 50
<detail>	Additional	O		N	
Type	<detail>	N		Y	Data type of the field. Required for additional fields. **1** String **2** Date **3** Boolean **4** Country code according to ISO 3166
Label	<detail>	S	100	Y	Readable label for the element
Value	<detail>	S	150	N	Initial field value
LegitimationDocument	[ROOT]	A		N	Individual legitimation documents in case defaults are not sufficient
Document	LegitimationDocument	A	50		The readable description shown to the agent / employee processing the order
eSignPDF	[ROOT]	O/A		N	Document(s) designated for signing, use array for multiple documents.
Data	eSignPDF	S		N	Base64 encoded PDF (PDF must be unprotected)
Filename	eSignPDF	S	50	N	Filename
SignaturePosX	eSignPDF	N		N	The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0. Position and field name cannot be used at the same time.
SignaturePosY	eSignPDF	N		N	
SignatureHeight	eSignPDF	N		N	

SignatureWidth	eSignPDF	N		N	Only applicable if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignaturePage	eSignPDF	N		N	Page to place the signature on
SignatureFieldName	eSignPDF	S		N	Use the given predefined signature field in the PDF. Position and field name cannot be used at the same time.
SignatureTag	eSignPDF	S		N	A tag (text), that is used to define the position of the signature (alterative to X/Y coordinates). It must only apear once in the document, otherwise an error will be returned.
SignatureOffsetPosX	eSignPDF	I		N	Can be used with SignatureTag
SignatureOffsetPosY	eSignPDF	I		N	Can be used with SignatureTag
Stamping	eSignPDF	A		N	Array of objects, used to place additional stamps on the document. This feature is only available when using Namirial TSP.
Page	Stamping	N		Dep	Page to place stamp on. Default is 1.
PosX	Stamping	N		Dep	Position of the stamp on the page. The units are PDF Points equivalent to 1/72 inch. Default is 100;100.
PosY	Stamping	N		Dep	
Tag	Stamping	S		N	A tag (text), that is used to define the position of stamps (alterative to X/Y coordinates). Stamp will be placed at each occurrence of matching text in the document.
OffsetPosX	eSignPDF	I		N	Can be used with Tag
OffsetPosY	eSignPDF	I		N	Can be used with Tag
Width	Stamping	N		N	Size of the stamp. The units are PDF Points equivalent to 1/72 inch. Default is 200;60.
Height	Stamping	N		N	

Content	Stamping	S	500	Dep	String content of the stamp, available placeholders are: **[DATE]** Date of signature **[TIME]** Time of signature **[DT]** Date and time of signature **[SIGNEE]** Name of signee
eSignHash	[ROOT]	O/A		N	
Hash	eSignHash	S		N	Base64 Hash of prepared PDF
Filename	eSignHash	S	50	N	Filename
eSignOptions	[ROOT]	O			
AdditionalCheckboxes	eSignOptions	A	300		Text for Checkboxes in the esign workflow that must be checked before signing is possible (Array of String(s))
MultiSignOptions	[ROOT]	O		N	
FirstOrderID	MultiSignOptions	S	14		To reference an eSign order to the master. Contract from first Order will be signed by each.
SignParameters	MultiSignOptions	O/A		N	Signature parameters for file(s) given with the first order
Filename	SignParameters	S	50	N	Filename given with first order
SignaturePosX	SignParameters	N		N	The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0. Default is 100;100.
SignaturePosY	SignParameters	N		N	
SignatureWidth	SignParameters	N		N	Only applicable if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch. Default is 200;60.
SignatureHeight	SignParameters	N		N	
SignaturePage	SignParameters	N		N	Page to place the signature on. Default is 1.
SignatureFieldName	SignParameters	S		N	Use the given predefined signature field in the PDF

IdentDataSigned	[ROOT]	S		N	Previously generated identification data for signing purposes
EmailNotifications	[ROOT]	O		N	Send final processing mails, overrides customer settings
OnPositiveIdent	EmailNotifications	S	150	N	Send a mail to this address in case of a final positive status
OnNegativeIdent	EmailNotifications	S	150	N	Send a mail to this address in case of a final negative status
OnTimeOut	EmailNotifications	S	150	N	Send a mail to this address in case of a final timeout status
OnLeave	EmailNotifications	S	150	N	Send a mail to this address in case the user leaves the identification process
Purpose	[ROOT]	S/O	150	N	Purpose of the identification that will be reported in the app to the end user. If a string is given here, it will be used for every language. In case language dependent strings are required, provide an object here and see the next line.
<lang>	Purpose	S	150	N	The key (<lang>) must be an ISO 3166-1 Alpha-2 language code, the value is the desired purpose text
eSignTanSmsText	[ROOT]	S		N	eSign TAN Text, only for eSign orders. Use [TAN] as a placeholder for the TAN.
verificationSmsText	[ROOT]	S		N	Mobile number verification SMS text. Use [TAN] as a placeholder for the TAN.
CustomerRouteTag	[ROOT]	S	250	Dep	Used to route orders to other preconfigured account(s), for example for invoicing purposes.
Autotest	[ROOT]	S	50	N	Used to activate testrobot flow (see 9.5).

3.1.5 Response Data

Element	Parent	Type	Len	Occ	Description
OrderID	[ROOT]	S	20	Y	Our internal ID of the Order, always unique
Ref	[ROOT]	S	100	Y	Your internal ID of the Order, can be set multiple times
TargetURL	[ROOT]	S	100	Dep	URL to redirect your customer
DeepLinkURL	[ROOT]	S	100	Dep	App DeepLink URL
AutoIdURL	[ROOT]	S	100	Dep	URL to autoID Web
SignedDataAccepted	[ROOT]	B		Dep	In case signed ident data has been provided, this flag will indicate if the data has been accepted
VideoHashes	[ROOT]	O		Dep	Video system reference data
ShortCode	VideoHashes	S	5	Dep	Reference number to start Video identification in identity App on iOS/Android
LongCode	VideoHashes	S	20	Dep	Reference number for identity Video SDK
UserID	[ROOT]	N		Dep	ID used for Shop user login, only provided in case this is an agent onboarding order
UserToken	[ROOT]	S	32	Dep	Token used for Shop user login, only provided in case this is an agent onboarding order
Error	[ROOT]	S		Dep	Error message in case an error occurred

3.1.6 HTTP Codes

202 Order accepted, JSON document in body
400 Bad request, error description in body
401 Authentication failed, please check username and password

3.1.7 Example Request


```
{
  "Ref": "Sign-me_Account_Creation_Test",
  "Product": 15,
  "Add": 8,
  "getTargetLink": true,
  "getDeepLink": true,
  "getVideoHashes": true,
  "WebhookURL": {
    "OnFinal": "https://webhook.site/1234567_890?OnFinal=[OrderID]"
  },
  "RedirectURL": {
    "Success":"https://www.example.com/redirect/positive?OrderID=[OrderID]",
    "Abort": "https://www.example.com/redirect/negative?OrderID=[OrderID]"
  },
  "Email": "e.mustermann@example.com ",
  "Mobile": "+491711234567",
  "Address": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Country": "DE"
  },
  "IdentData": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Birthdate": "1964-08-12",
    "Birthplace": "Berlin",
    "Additional": {
      "idCardDateOfIssue": {
        "Type": 2,
        "Label": "Date of Issue",
        "Value": "2010-01-11"
      }
    }
  }
}
```


3.1.8 Example Response


```
{
    "OrderID": "99921269855041",
    "Ref": "Sign-me_Account_Creation_Test",
    "TargetURL": "https://www.identity.tm/status/37C4B60A72A1C6C74FC038282CED500B",
    "DeepLinkURL": "https://www.identity.tm/ident/37C4B60A72A1C6C74FC038282CED500B",
    "VideoHashes": {
        "ShortCode": "SHU8T",
        "LongCode": "a9PPz-eFcNBifsk--bNJ"
    }
}
```


3.2 Checking the status of orders

When an order has been processed, it is of course necessary to check the status of the order. This usually happens after the `OnFinal` webhook has been sent.

Please do not try to query the order status and result data synchronously from the webhook itself. To keep the system responsive, it is expected that the call completes under 5 seconds.

3.2.1 Final status codes vs. non-final status codes

A status code can be *final* or *non-final*. *Final* in this case means, that the process has come to a natural end and the user will not be able to try it again using the same order. In case of a positive identification / signing, result data will be available (with a special exception in case you requested us to sign ident PDF documents, these will only be available after status 80 has been set).

Since eSign and identification orders work differently, a different set of status codes is considered final.

3.2.2 Extended status codes vs. normal status codes

Each status code is either a normal one or an extended one. Normal (or non-extended) ones are used to inform you about the major steps an order took while being processed.

Extended status codes are used to provide you with a more detailed and more in-depth view of how the order has been process by us.

The default variant of `getStatus` will only return normal status codes. If you want to receive the extended ones as well, you need to use the `ExtendedList` variant of `getStatus`. The normal status codes will be included as well.

3.2.3 List of status codes

Please take note, that not all codes are relevant for every product. For example, most codes above 10000 are specific to autoID.

The status codes 29 (uncorrectable error in external systems) and 139 (correctable error in external systems) are especially relevant when you intend to implement a customer-side eSign processing according to chapter 5.

Both errors refer to errors in external systems, here namely the systems of the TSP.

If you encounter status code 29 in such a process, there will be no way to finish the order and a new order needs to be created. Code 139 indicates a possibly correctable error.

Kind	Description / Meaning	Final non-eSign	Final eSign	Extended
6	Identification positively conducted	Y	N	N
7	E-mail has been sent	N	N	Y
11	Order is being processed by QM	N	N	N
12	Customer could not be reached by phone	N	N	N
14	Document sent to client	N	N	N
16	Identification refused by customer	Y	Y	N
17	Identification negatively conducted	Y	Y	N
21	The processing was aborted	N	N	N
22	The processing was terminated	Y	Y	N
23	Document verified	N	N	Y
24	Document erroneous in postprocessing	N	N	Y
25	Order initialized	N	N	Y
29	Uncorrectable error in external system	N	N	Y
30	Order has been processed by QM	N	N	N
49	eID IDapp link created	N	N	Y
50	eID IDapp request	N	N	Y
51	eID data gathered	N	N	Y
53	eID aborted/not possible	N	N	Y
54	Support	N	N	Y
55	Customer could not be reached by phone	N	N	N
56	Phone number is incorrect	N	N	N
57	Identification not possible, customer was repeatedly not present at appointments	N	N	N
58	Identification refused by customer by phone	Y	Y	N
63	Cancellation	Y	Y	N
68	Change of procedure to Shop	N	N	N
70	Spoken on answering machine	N	N	Y

71	SMS sent	N	N	Y
73	Change of procedure to Video	N	N	N
74	Change of procedure to eiD	N	N	Y
75	Video identification conducted	N	N	N
77	Video identification in waiting room	N	N	Y
78	Video identification call has begun	N	N	Y
79	Video identification call has been left	N	N	Y
80	Identification data ready for pick-up	N	N	N
83	Document signature request	N	N	Y
84	Document signed	N	Y	Y
87	Request for new Order	N	N	N
89	Possible fraud attempt by customer	N	N	Y
94	eSign Account created	N	N	Y
95	eSign Account verified	N	N	Y
96	eSign Account already exists	N	N	Y
98	Change of procedure to autoID	N	N	N
99	TimeOut	Y	Y	N
100	TimeOut after identification	Y	Y	N
101	eSign documents verified and hashes generated	N	N	Y
102	eSign document preview	N	N	Y
103	eSign aborted	N	Y	N
104	Video ident waiting for verification	N	N	Y
105	Ident is under verification	N	N	Y
106	Video Ident verification positive	N	N	Y
107	Video Ident verification data error found	N	N	Y
108	Video Ident verification negative	N	N	Y
109	autoID process not possible	N	N	Y
139	Correctable error in external system, please try	N	N	Y
153	autoId abort	N	N	Y
10000	no address verification	N	N	Y
10001	no additional document	N	N	Y

10002	customer without identification document	N	N	Y
10003	photo resolution too low to read text	N	N	Y
10004	customer not present	N	N	Y
10005	customer less than minimum age	N	N	Y
10006	poor lighting conditions	N	N	Y
10007	identification document unrecognisable	N	N	Y
10008	Other cancellation reasons	N	N	Y
10009	identical selfies or images	N	N	Y
10010	Paper driving licence not all pages visible in the photo	N	N	Y
20000	Photo in document does not match person	N	N	Y
20001	Document looks forged	N	N	Y
20002	Repetitive fraud attempt	N	N	Y
20003	Fraud attempt	N	N	Y
20004	Negative according to test criteria	N	N	Y
20005	Document not supported	N	N	Y
20006	Document expired	N	N	Y
20007	Document damaged	N	N	Y
20008	Date of issue Driving licence class B not readable	N	N	Y
20009	Driving licence expired	N	N	Y
20010	Driving licence not supported	N	N	Y
20011	Driving licence damaged	N	N	Y

3.2.3.1 Sub-Status Codes

For some specific status codes, a more detailed view is available. The details are only included in the response if the appropriate variant of the method is called.

Keep in mind that not all status codes have sub-status codes and even status codes that have sub-status codes might not always carry a sub-status.

The list of sub-status codes might grow at any point in time, so do not rely on the list being exhaustive.

Kind	SubKind	Description / Meaning

6	19	Order with signed ident data
7	10	User left video call
	12	Invitation to identification
	21	Temporary cancellation / technical abort with retry option
	24	Reminder
	25	Final message
	28	TAN
	33	Shop invitation
	37	Signature re-entry link
	55	Identification appointment
	58	eSign reminder
17	43	Document not supported
	48	Identification criteria
	61	TAN tried too often
	72	Identification document photo does not match person
	73	Identification document invalid, expired, non-existent, or not permitted
	74	Address confirmation document invalid, expired, non-existent, or not permitted
	75	Driver’s license invalid, expired, non-existent, or not permitted
21	14	Unable to establish video connection to user
	15	Resolution too low to conduct video identification
	18	Unable to establish audio connection to user
	22	User not present
	27	Unsupported identification document
	29	Identification data does not match
	32	User without identification document
	34	User speaks no supported language
	36	Other reasons
	38	Identification document damaged
	39	Lighting conditions too poor to conduct identification
	42	Identification document expired
	47	Negative due to identification criteria

	50	User unable to show any document
	53	Type of identification document could not be determined
	54	No address verification possible
23	20	Order with signed ident data
	41	As configured
29	63	The certificate associated with the key has expired
	64	Existing Holder with different person data
	65	Mobile number already assigned to another holder with a certificate that is still valid and has not revoked
	76	Mobile number already assigned to another holder till a specific date
58	30	User is no longer interested in product
71	16	TAN
	17	Namirial TAN
	23	autoID
	44	Sign-Me password
	46	Reference
	49	AES TAN
	56	eSign re-entry TAN
	62	eID TAN
103	79	PDF hashing failed
139	66	Sending OTP TAN failed
153	176	Document is blurry and mandatory data cannot be read due to the blur.
	177	Document has glare and mandatory data cannot be read due to the glare.
	178	Pictures of the document are dark and it is not possible to read the mandatory data or verify the authenticity of the document.
	179	Mandatory data is covered by the user while taking the picture.
	180	Document is positioned at such an angle that mandatory data cannot be read or document cannot be verified.
	181	Any other reason due to which mandatory data cannot be read.
	182	Document used during the identification is not supported for the customer's use case.
	183	Document used during the identification is expired.
	184	Wrong side of the document is scanned during the process.

	185	Document is worn out. Either data cannot be read out or the document cannot be verified.
	186	Document has such stickers which are not acceptable and the document used is considered as damaged document.
	187	Document has text written over it which makes the document not readable or not verifiable. If the sticker is legit one and added by the authorities while issuing the document then the document will be acceptable and not cancelled due to this reason..
	188	Document used during the identification is broken.
	189	Any other reason for a damaged document.
	190	Security features of the document are not visible because user did not move the document correctly.
	191	Security feature video is too short to detect if there are holograms in the document.
	192	Security feature video cannot be played for the agent to review holograms.
	193	Any other issues with the security feature video.
	194	Photo quality of the additional document in the process is not acceptable.
	195	Additional document used in the identification process is severely outworn, written or drawn on, ripped or broken.
	196	Addiitonal document used in the identification process is an expired document.
	197	Additional document like Drivers License is missing in the identification process but it was required.
	198	Any other issues with the additional document used in the identification process.
	199	Any other issues with the document used in the identification process.
	200	Customer needs proof of address from the user as the additional document but user did not provide it in the identification process.
	201	User has covered the face during the face comparison process unintentionally like wearing the face mask.
	202	Selfie taken by the user is blurry and cannot be used to compare the face with the identification document.
	203	Photo of the user on the ID document has glares and selfie cannot be compared with it.
	204	Selfie taken by the user is too dark to compare the face of the person with the photo on the identification document.
	205	Selfie taken by the user is on such an angle that it is not possible to compare it with the photo on the identification document.

	206	Any other issues with the selfie which restrict ident sepcialist to compare the selfie of the user with the photo on the identification document.
	207	Due to a technical reason, ident specialist cannot finish the identity verification process.
	208	Due to a technical reason, ident specialist cannot see the data submitted by the user in the identification process.
	209	Any other reason due to which the identification process cannot be completed by the ident specialist.

3.2.4 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getStatus/OrderID
[GET] https://customer.identity.tm/api/2.10/getStatus/OrderID/IncludeSubKind
[GET] https://customer.identity.tm/api/2.10/getStatus/OrderID/ExtendedList
[GET] https://customer.identity.tm/api/2.10/getStatus/OrderID/ExtendedList/IncludeSubKind
```

3.2.5 Variants

Variant	Description
Default	Return only normal status codes
IncludeSubKind	Also return sub-status codes
ExtendedList	Also return extended status codes

3.2.6 Response Data

Element	Parent	Type	Len	Occ	Description
OrderID	[ROOT]	S	20	Y	Requested OrderID
Ref	[ROOT]	S	100	Y	Corresponding Ref
Executed	[ROOT]	DTS		Y	Timestamp at which the request has been executed
Status	[ROOT]	A		Y	Array of status codes
Kind	Status	N		Y	Status code
SubKind	Status	N		Dep	Only when `/IncludeSubKind` variant of the method is used
Time	Status	DTS		Y	Timestamp at which status has been set
Text	Status	S		Dep	Explanatory text

3.2.7 HTTP Codes

200 OK, JSON document in body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.2.8 Example Request

```
[GET] https://customer.identity.tm/api/2.0/getStatus/99921269855041/ExtendedList
```

3.2.9 Example Response


```
{
    "OrderID": "99921269855041",
    "Ref": "Sign-me_Account_Creation_Test",
    "Executed": "2018-09-05T11:06:21+02:00",
    "Status": [
        {
            "Kind": 25,
            "Time": "2018-09-05T10:53:59+02:00"
        },
        {
            "Kind": 71,
            "Time": "2018-09-05T10:54:00+02:00",
            "Text": "signme Passwort"
        },
        {
            "Kind": 94,
            "Time": "2018-09-05T10:54:00+02:00"
        },
        {
            "Kind": 77,
            "Time": "2018-09-05T10:56:37+02:00"
        },
        {
            "Kind": 78,
            "Time": "2018-09-05T10:56:45+02:00"
        },
        {
            "Kind": 71,
            "Time": "2018-09-05T10:59:11+02:00",
            "Text": "TAN"
        },
        {
            "Kind": 75,
            "Time": "2018-09-05T11:06:21+02:00"
        },
        {
            "Kind": 6,
            "Time": "2018-09-05T11:06:21+02:00"
        },
        {
            "Kind": 23,
            "Time": "2018-09-05T11:06:21+02:00"
        },
        {
            "Kind": 95,
            "Time": "2018-09-05T11:09:51+02:00"
        }
    ]
}
```

3.3 Retrieving identification results in JSON format

The usual next step after an `OnFinal` webhook and the `getStatus` request is the `getIdentData` request with the subsequent logic.

If required, your account can be configured to return the data encrypted.

3.3.1 Requirements

The order must be finally processed. For a list of final status see chapter 3.2.3.

3.3.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/IncludeInitialData
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/IncludeIdentifyMethod
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/IncludeIdentifier
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/Signed
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/crypt
```

3.3.3 Variants

Variant	Description
Default	Normal variant, response as described below
IncludeInitialData	Include initial data as given to the putOrder request
IncludeIdentifyMethod	Include an additional element in the response “IdentifyMethod”
IncludeIdentifier	Include the internal user ID of the identifier
Signed	Return signed and encrypted version of the identification data for later re-use
crypt	Encrypt response content using AES-256/GCM/NoPadding and a 23 byte long IV prefixed to the encrypted content. This variant can be used in combination with the other variants.
IncludeEidJwt	Include transaction attestation for the identitfication by eID (JWT)

3.3.4 Response Data

Note: To include the images in the response, your account needs to be configured accordingly. Contact IDnow to change your configuration. This setting has an impact on the data format – if images are included, the element `IdentData` will be an object otherwise an array.

Element	Parent	Type	Len	Occ	Description

Identitfied	[ROOT]	DTS		Y	Time of identification The typo in the name is kept for backward compatibility
IdentifyMethod	[ROOT]	S		Dep	Method of identification used for this order; one of: • `Shop` • `Video` • `eID` • `Giro` • `AutoID` • `SmartID` *Only if specifically requested*
Identifier	[ROOT]	S / N		Dep	Internal user ID of user who conducted the identification *Only if specifically requested*
UserIp	[ROOT]	S		Dep	IP address of user while conducting the identification. Will only be collected and returned, if configured on your account.
OcrStatus	[ROOT]	N		Dep	In case of POS ident orders, this field tells about OCR processing status: **0** no OCR processed **1** OCR and manual images processed **2** only OCR processed
eIdUsed	[ROOT]	B		Dep	In case of POS ident orders, this field tells if eID has been conducted or not
SecurityCheckFailed	[ROOT]	B		Dep	In case of POS ident orders, this field is true in case the on-device security check failed
FraudAttempt	[ROOT]	B		Dep	`True` indicates a potential fraud attempt. *Only if fraud attempt has been flagged by an agent.*

eidJwt	[ROOT]	S		Dep	Transaction attestation for the identitfication by eID (JWT) *Only if specifically requested*
IdentData	IdentData	O/A		Y	Type (object or array) depends on whether images are included in response or not
Type	IdentData	N		Dep	**1** Checked data **2** Initial data from putOrder
Firstname	IdentData	S	50	Y	
Surname	IdentData	S	50	Y	
Street	IdentData	S	100	Y	
ZipCode	IdentData	S	5	Y	
City	IdentData	S	100	Y	
Country	IdentData	S		Dep	Only if noted on identity document
Birthdate	IdentData	DS		Y	
Birthplace	IdentData	S	100	Y	
Birthname	IdentData	S	75	Dep	Only if requested
Nationality	IdentData	S	50	Dep	Only if noted on identity document
Title	IdentData	S	50	Dep	Only if noted on identity document
Artistname	IdentData	S	50	Dep	Only if noted on identity document
Gender	IdentData	S		Dep	female/male/diverse
Mobile	IdentData	S	50	Dep	Only if verified with TAN
Email	IdentData	S	100	Dep	Only if verified with TAN
IDCard	IdentData	O		Dep	In case of a license check without ID, this field will not be included
CardNo	IDCard	S	50	Dep	In case of an eID-only identification, will field will not be included

eidRestrictedIdentifier	IDCard	S		Dep	Restricted Identifier (dienste- und kartenspezifisches Kennzeichen - DKK); only if eID was used to identity user
extDocumentType	IDCard	N		Dep	Specifies the document(s) used. If multiple documents have been used, the individual values will be or'ed bitwisely. **0** Custom document **1** ID Card **2** Passport **4** Residence permit **8** Driver's license **16** German eID-Card for EU citizens All other values are reserved for future use. New values may be added at any time.
PlaceOfIssue	IDCard	S	100	Dep	In case of an eID-only identification, will field will not be included
CountryOfIssue	IDCard	S	3	Dep	Depends on customer specific requirements and document
DateOfIssue	IDCard	DS		Dep	In case of an eID-only identification, will field will not be included
ValidUntil	IDCard	DS		Dep	Depends on customer specific requirements
Type	IDCard	S		Dep	IDCard/Passport or LegitimationDocument provided in putOrder
DriverLicence	IdentData	O		Dep	If enabled in putOrder
LicenceNo	DriverLicence	S	50	Dep	If enabled in putOrder
Classes	DriverLicence	S	50	Dep	
DateOfIssue	DriverLicence	DS		Dep	
Additional	IdentData			Dep	If set in putOrder

<detail>	Additional	S	150	N	If set in putOrder
Images	IdentData	A		Dep	If enabled on your account
Image	Images	S			Base64 encoded Data
Filename	Images	S	100		Filename for the data

3.3.5 HTTP Codes

200 OK, JSON document in response body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.3.6 Example Request

```
[GET] https://customer.identity.tm/api/2.10/getIdentData/99921269855041/IncludeInitialData
```

3.3.7 Example Response


```
{
    "Identitfied": "2018-09-05T11:06:21+02:00",
    "IdentData": [
        {
            "Type": 1,
            "Firstname": "Erika",
            "Surname": "Mustermann",
            "Street": "Heidestr. 17",
            "ZipCode": "43000",
            "City": "Köln",
            "Birthdate": "1964-08-12",
            "Birthplace": "Berlin",
            "Nationality": "DE",
            "Gender": "female",
            "Mobile": "+4915114084205",
            "IDCard": {
                "CardNo": "T22000129",
                "PlaceOfIssue": "Stadt Köln",
                "ValidUntil": "2020-10-31",
                "Type": "IDCard"
            }
        },
        {
            "Type": 2,
            "Firstname": "Erika",
            "Surname": "Mustermann",
            "Street": "Heidestr. 17",
            "ZipCode": "43000",
            "City": "Köln",
            "Birthdate": "1964-08-12",
            "Birthplace": "Berlin",
            "Nationality": "DE",
            "IDCard": {
                "CardNo": null,
                "PlaceOfIssue": null,
                "ValidUntil": "1970-01-01"
            }
        }
```

```
    ]
}
```

3.4 Retrieving identification results as a PDF

The same `getIdentData` logic as described in chapter 3.3 applies here too. The content is basically the same except that a PDF will be returned, and all images/screenshots are included by default.

3.4.1 Requirements

Same as `getIdentData`, additionally this endpoint must be enabled by IDnow.

3.4.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getIdentDataPDF/OrderID
[GET] https://customer.identity.tm/api/2.10/getIdentDataPDF/OrderID/crypt
```

3.4.3 Variants

Variant	Description
Default	Normal variant, response as described below
crypt	Encrypt response content using AES-256/GCM/NoPadding and a 23 byte long IV prefixed to the encrypted content. This variant can be used in combination with the other variants.

3.4.4 HTTP Codes

200 OK, PDF in body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.4.5 Example Request

```
[GET] https://customer.identity.tm/api/2.10/getIdentDataPDF/99921269855041
```

3.4.6 Example Response

Resultat
IDENTprüfung

Auftrag: 99921267754340 Referenz: Shop GWG

Angabe	Geliefert	Geprüft
Nachname	Mustermann	Mustermann
Vorname	Erika	Erika
Strasse	Heidestr. 17	Heidestr. 17
PLZ	51147	51147
Ort	Köln	Köln
Geburtsdatum	12.08.1964	12.08.1964
Geburtsort	Berlin	Berlin
Nationalität		DE
Künstlername		
Titel		
Ausweisnummer		T22000129
AusstellungsLand		
AusstellungsOrt		Stadt Köln
Gültig bis		31.10.2020

Lichtbild und Merkmale gemäß Ausweisdokument mit der zu identifizierenden Person verglichen und Identität festgestellt.

Resultat positiv am 03.09.2018 08:52
Identifizierer
Prüfer 1872

06.09.2018 16:10 (1/3)

3.5 Retrieving signed documents

Endpoint for fetching processed/signed eSign PDF documents

3.5.1 Requirements

`eSignPDF` element was set in `putOrder` and final status is 84 (see chapter 3.2.3 for status details).

3.5.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getESignPDF/OrderID
[GET] https://customer.identity.tm/api/2.10/getESignPDF/OrderID/crypt
```

3.5.3 Variants

Variant	Description
Default	Normal variant, response as described below
crypt	Encrypt response content using AES-256/GCM/NoPadding and a 23 byte long IV prefixed to the encrypted content. This variant can be used in combination with the other variants.

3.5.4 Response Data

Element	Parent	Type	Len	Occ	Description
PDFs	[ROOT]	A		Y	Array of documents
Base64Data	PDFs	S		Y	Base64 encoded PDF data

3.5.5 HTTP Codes

200 OK, JSON document in response body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.6 Retrieving signing hashes

Endpoint for fetching processed/signed eSign hashes.

3.6.1 Requirements

`eSignHash` element was set in `putOrder` and final status is 84.

3.6.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getESignHash/OrderID
```

3.6.3 Response Data

Element	Parent	Type	Len	Occ	Description
Hashes	[ROOT]	A		Y	Array of hashes
<Name>	Hashes	S		Y	Hash value, <Name> was given in `putOrder` request

3.6.4 HTTP Codes

200 OK, JSON document in body
401 Authentication failed, please check username and password
400 Bad request, error description in body
404 Order not found

3.7 Retrieving AES audit log file

Endpoint for fetching a PDF-based audit log file about the signing process. Only available for AES orders.

3.7.1 Requirements

Order is an AES order and final status is 84.

3.7.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getESignAuditLogPDF/OrderID
```

3.7.3 Response Data

Binary PDF data in body.

3.7.4 HTTP Codes

200 OK, PDF document in body
401 Authentication failed, please check username and password
400 Bad request, error description in body
404 Order not found

3.8 Retrieving recorded images

Usually, recorded images will be provided along the ident data. But in some cases, it might be useful to split that up and request the images separately.

3.8.1 Requirements

The order must be finally processed. For a list of final status see chapter 3.2.3.

3.8.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getImages/OrderID
```

3.8.3 Response Data

Element	Parent	Type	Len	Occ	Description
Images	[ROOT]	A		Dep	Only if images were recorded
Image	Images	S			Base64 encoded Data
Filename	Images	S	100		Filename for the data

3.8.4 HTTP Codes

200 OK, JSON document in response body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.8.5 Example Request

```
[GET] https://customer.identity.tm/api/2.10/getImages/99921269855041
```

3.8.6 Example Response

```
{
    "Images": [
        {
            "Image": "/9j/4AAQSkZJRgABAQAASABIAAD/...",
            "Filename": "back_20211008-091705-123.jpg"
        },
        {
            "Image": "/9j/4AAQSkZJRgABAQAASABIAAD/...",
            "Filename": "front_20211008-091623-123.jpg"
        }
    ]
}
```

3.9 Requesting video file provisioning

In previous versions of our API, one could download a video file of the identification process immediately after retrieving the identification data.

This behavior is no longer recommended, as specific products to not support that. It is therefore necessary to request video file provisioning with a call to this function first.

Once the video file is available, the given callback URL will be called with a GET request to indicate video file availability.

3.9.1 Requirements

The desired products must be associated with and enabled on your account by IDnow.

3.9.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/getVideoFileBinaryAsync/OrderID
```

3.9.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
callbackUrl	[ROOT]	S		Y	URL to call once video is available

3.9.4 HTTP Codes

200 OK
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.10 Retrieving video binary files after provision

This file is provided, when identification was conducted in video with a recording of the session from the agent's point of view.

3.10.1 Requirements

This file is provided, when final status from video is 6 or 89 and the recording requirement was set either in the Add element of the `putOrder` request or globally configured in the account for legal (e.g. German AML) compliance by IDnow.

Furthermore, if you did not request provisioning of the file or did not wait for the callback to happen before calling this function, you might get a placeholder video instead.

3.10.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getVideoFileBinary/OrderID
[GET] https://customer.identity.tm/api/2.10/getVideoFileBinary/OrderID/crypt
```

3.10.3 Variants

Variant	Description
Default	Normal variant, response as described
crypt	Encrypt response content using AES-256/GCM/NoPadding and a 23 byte long IV prefixed to the encrypted content. This variant can be used in combination with the other variants.

3.10.4 HTTP Codes

200 OK, binary video file (video/mp4 or video/webm) in body
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found

3.11 Retrieving audio binary files

In some countries it is sufficient, to have an audio recording instead of a video file. In these cases, the configuration will be made by IDnow for the given account. An audio recording cannot be requested with the putOrder request.

3.11.1 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getVoiceFiles/OrderID
[GET] https://customer.identity.tm/api/2.10/getVoiceFiles/OrderID/crypt
```

3.11.2 Variants

Variant	Description
Default	Normal variant, response as described
crypt	Encrypt response content using AES-256/GCM/NoPadding and a 23 byte long IV prefixed to the encrypted content. This variant can be used in combination with the other variants.

3.11.3 HTTP Codes

200 OK, JSON document in body
401 Authentication fails, please check username and password
400 Bad request, error description in body
404 Order not found

3.11.4 Example Request

```
[GET] https://customer.identity.tm/api/2.10/getVoiceFiles/99920006C38842
```

3.11.5 Example Response

```
{
  "Voicefiles": [
    { "Filename":"sound_4348.mp3","Base64Data":"\/\/uQBAAAA......." }
  ]
}
```

3.12 Deleting identification data

By default, IDnow will delete all result files 7 days after fulfilment. Nevertheless, it would be best, to delete the result files after they have been collected completely and successfully.

Some confusion exists on when to use `delIdentData` and when to use `cancelOrder`. As a rule of thumb, use `cancelOrder` before the `OnFinal` webhook has been triggered and `delIdentData` after that point.

3.12.1 Method and URL

```
[DELETE] https://customer.identity.tm/api/2.10/delIdentdata/OrderID
```

3.12.2 HTTP Codes

202	Accepted, data deleted
400	Bad request, error description in body
401	Authentication failed, please check username and password
404	Order not found or already deleted
406	Invalid order state

3.12.3 Example Request

```
[DELETE] https://customer.identity.tm/api/2.10/delIdentdata/99920006C38842
```

3.13 Retrieving OrderIDs for given Ref

Gives you an Array of matching OrderIDs (of last 90 days). Only allowed/possible for Refs with Characters of:

```
a-z A-Z 0-9 - . +
```

3.13.1 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getOrdersByRef/Ref
```

3.13.2 HTTP Codes

200	OK, JSON document in body
400	Client Error
401	Authentication fails, please check username and password
400	Bad request, error description in body
404	No Order(s) found

3.13.3 Example Request

```
[GET] https://customer.identity.tm/api/2.10/getOrdersByRef/543210
```

3.13.4 Example Response

```
{
 [
    "99921801997040",
    "99921800019939"
 ]
}
```

4 Working with incomplete Orders

4.1 Setting up a Time Out

Please contact itsupport@identity.tm to set up a general time out for incomplete orders.

4.2 Cancelling orders

If an open order shall not be processed anymore, you can cancel it with this endpoint. The order data will not be deleted, but the user will not be able to proceed with his identification or signature.

4.2.1 URL

```
[POST] https://customer.identity.tm/api/2.10/cancelOrder/OrderID
```

4.2.2 HTTP Codes

202	Accepted
400	Bad request, error information in body
401	Authentication failed
404	Order not found
406	Order not in a cancellable state

4.2.3 Example Request

```
[POST] https://customer.identity.tm/api/2.10/cancelOrder/99920006C38842
```

5 Customer-side eSign processing

All these operations need to be enabled for your account. They work on previously created orders. See chapter 3.1 for details.

If you desire a deeper integration of the signing process into your corporate design and want to provide the necessary function on your side, these functions will enable to you carry out the whole eSign process.

5.1 Process order and start signing

Use this function to start the signing process. A TAN will be sent to the signee.

5.1.1 Requirements

This function must be enabled for your account.

5.1.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/requestSign/OrderID
```

5.1.3 HTTP Codes

202	Accepted
400	Bad request, error description in body
401	Authentication failed, please check username and password
404	Order not found or already deleted
406	Invalid order state
410	Order has already been processed

5.1.4 Example Request

```
[POST] https://customer.identity.tm/api/2.10/requestSign/99920006C38842
```

5.2 Confirm signing

Sign the document. To do so, the TAN previously sent need to be provided.

5.2.1 Requirements

This function must be enabled for your account.

5.2.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/confirmSign/OrderID
```

5.2.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
tan	[ROOT]	S	6	Y	The TAN the user received.

5.2.4 HTTP Codes

202 Accepted
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found or already deleted
409 Wrong TAN given
410 Order has already been processed
412 Signing process has not been started, call `requestSign` first
429 Too many tries with a wrong TAN, order cancelled

5.2.5 Example Request

```
[POST] https://customer.identity.tm/api/2.10/confirmSign/99920006C38842

{
    "tan": "123456"
}
```

5.3 Request a new TAN

In case the TAN has not been received by the signee, you can request to send a new TAN to the signee.

5.3.1 Requirements

This function must be enabled for your account.

5.3.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/requestResendSignTan/OrderID
```

5.3.3 HTTP Codes

202	Accepted
400	Bad request, error description in body
401	Authentication failed, please check username and password
404	Order not found or already deleted
410	Order has already been processed
412	Signing process has not been started, call `requestSign` first

5.3.4 Example Request

```
[POST] https://customer.identity.tm/api/2.10/requestResendSignTan/99920006C38842
```

6 Mobile Phone Number Verification

All these operations need to be enabled for your account. They work on previously created orders. See chapter 3.1 for details.

In case you set up a deeply integrated PoS identification and possibly signing, it will be necessary to verify the user's control over a mobile device identified by a mobile phone number. Instead of implementing that completely on your own, you can use our services to make handling easier.

6.1 Process order and start verification

Use this function to start verification process. A TAN will be sent to the user.

6.1.1 Requirements

This function must be enabled for your account.

6.1.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/requestPhoneVerification/OrderID
```

6.1.3 HTTP Codes

202	Accepted
400	Bad request, error description in body
401	Authentication failed, please check username and password
404	Order not found or already deleted
406	Invalid order state
410	Order has already been processed

6.1.4 Example Request

```
[POST] https://customer.identity.tm/api/2.10/requestPhoneVerification/99920006C38842
```

6.2 Prove control over mobile phone

Prove, that the user has control over the given mobile phone number. To do so, the TAN previously sent need to be provided.

6.2.1 Requirements

This function must be enabled for your account.

6.2.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/provePhoneControl/OrderID
```

6.2.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
tan	[ROOT]	S	5	Y	The TAN the user received.

6.2.4 HTTP Codes

202 Accepted
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found or already deleted
409 Wrong TAN given
410 Order has already been processed
412 Process has not been started, call `requestPhoneVerification` first
429 Too many tries with a wrong TAN, order cancelled

6.2.5 Example Request

```
[POST] https://customer.identity.tm/api/2.10/provePhoneControl/99920006C38842

{
    "tan": "123456"
}
```

6.3 Request a new TAN

In case the TAN has not been received by the signee, you can request to send a new TAN to the signee.

6.3.1 Requirements

This function must be enabled for your account.

6.3.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/requestResendPhoneTan/OrderID
```

6.3.3 HTTP Codes

202	Accepted
400	Bad request, error description in body
401	Authentication failed, please check username and password
404	Order not found or already deleted
410	Order has already been processed
412	Process has not been started, call `requestPhoneVerification` first

6.3.4 Example Request

```
[POST] https://customer.identity.tm/api/2.10/requestResendPhoneTan/99920006C38842
```

7 Re-using identifications to sign documents

If your account is eSign-enabled and is using Namirial, it is possible to store identification data and re-use it later to avoid a going through a whole new identification process.

To be able to do so, a two-step approach is used:

1. Retrieve and store a signed version of the identification data
2. Provide this identification data when placing a new order for the same user

To retrieve the signed version of the identification data you need to call the “Signed”-variant of `getIdentData` (`getIdentData/`**`OrderID`**`/Signed`) as described in chapter 3.3 and store the result body.

When you now want to sign a document and re-use the identification, you provide the stored data in the element `IdentDataSigned` inside the body of the `putOrder` call.

Before re-using the given data, it will be checked, if

- It has not been tampered and is signed correctly,
- The given data is sufficient to be used for signing purposes,
- The identity document used for the identification is still valid.

In case the data is considered valid for signing purposes, the user will immediately be redirected to the signing process without a further identification, otherwise the user will be re-identified.

8 Service Monitoring

8.1 Checking System Liveness

To check the liveness of our service, you can call a special function without side effects. This function only checks the basic server availability.

8.1.1 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/serverStatus
```

8.1.2 HTTP Codes

200	Service available
500	Service not available
None	Service not available

8.2 Checking System Status

The system continually monitors its own status. To query information, you can call this function at any time.

Please note, that the availability of mailing only reflects our internal mailing system. If you gave us an external mailing service to use, this service will not be monitored for you.

Furthermore, information about AES will only be available if your account has been configured to provide AES functionality.

8.2.1 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/systemStatus
```

8.2.2 Response Data

Element	Parent	Type	Len	Occ	Description
serverAlive	[ROOT]	B		Y	Servers are running
apiAvailable	[ROOT]	B		Y	Customer API working
videoAvailable	[ROOT]	B		Y	Video system working
mailingAvailable	[ROOT]	B		Y	Mail-System available
smsTanAvailable	[ROOT]	B		Y	SMS TAN working
autoIdentAvailable	[ROOT]	B		Dep	IDnow AutoIdent servers alive Only if enabled
assureIdAvailable	[ROOT]	B		Dep	AssureID servers alive Only if enabled
readIdAvailable	[ROOT]	B		Dep	ReadID servers alive Only if enabled
iProovAvailable	[ROOT]	B		Dep	iProov servers alive Only if enabled
eSign	[ROOT]	O		Dep	Only if eSign is enabled
serverAlive	eSign	B		Dep	QES servers alive
qesAvailable	eSign	B		Dep	QES systems working
fesServerAlive	eSign	B		Dep	AES servers alive
fesAvailable	eSign	B		Dep	AES systems working

8.2.4 HTTP Codes

200	OK, JSON document in body
401	Authentication failed, please check username and password
500	Services not available at all
None	Services not available at all

8.2.5 Example Request

```
[GET] https://customer.identity.tm/api/2.10/systemStatus
```

8.2.6 Example Response

```
{
    "serverAlive": true,
    "apiAvailable": true,
    "videoAvailable": true,
    "mailingAvailable": true,
    "smsTanAvailable": true,
    "eSign": {
            "serverAlive": true,
            "qesAvailable": true,
            "fesServerAlive": true,
            "fesAvailable": true
    }
}
```

9 Additional Functions

9.1 Checking sign-me User Status

This method is only applicable if you wish to create an eSign order for sign-me (D-Trust). Before creating the order, you can check the user status in the sign-me system to find out if the given user already has a sign-me account and if he or she is able to sign with a qualified electronic signature (QES). Both elements in the response need to be true in order to create a plain signature Order (Add = 16 in `putOrder`). Otherwise, the account needs to be created and verified (Add = 0 *or* 8 in `putOrder` – The Add value 0 represents the default setting with creation/verification of the account plus PDF document(s) that need(s) to be signed – with the Add value 8 it is the same except for the PDF and signature part).

9.1.1 Requirements

Your account must be configured for eSign using sign-me.

9.1.2 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/checkSignmeUser
```

9.1.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
Email	[ROOT]	String		Y	The sign-me username (e-mail)
signatureType	[ROOT]	String	3	N	Optionally check if the user is able to sign with a specific signature type. Possible values: **QES** QES **ADV** AES **BAS** Basic electronic signature/BES

9.1.4 Response Data

Element	Parent	Type	Len	Occ	Description
UserExists	[ROOT]	B		Y	The sign-me account exists
UserIsAbelToSign	[ROOT]	B		Y	The sign-me account is qualified and able to sign

9.1.5 HTTP Codes

200 OK, JSON document in body
400 Bad request, error description in body
401 Authentication failed, please check username and password

9.1.6 Example Request

```
[POST] https://customer.identity.tm/api/2.10/checkSignmeUser

{
  "Email": "e.mustermann@example.com"
}
```

9.1.7 Example Response

```
{
    "UserExists": false,
    "UserIsAbleToSign": false
}
```

9.2 Changing the Customer API Password Programmatically

It is possible to request a password change programmatically. To be able to do so, you need to provide a secure mobile phone number to IDnow's Service Desk first to enable that functionality.

The password change process consists of two steps, requesting a password change and confirming it. The pre-defined mobile number needs to be provided in the first step as an additional secret.

A TAN number will be sent via SMS to the mobile phone. This TAN needs to be included in the confirmation request.

9.2.1 Request Password Change

9.2.1.1 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/requestNewPassword
```

9.2.1.2 JSON Request Body

Element	Parent	Type	Len	Req	Description
mobile	[ROOT]	PN	64	Y	Mobile phone number to send confirmation TAN to

9.2.1.3 HTTP Codes

200 OK, TAN is sent to the mobile number
400 Bad request, error description in body
401 Authentication failed, please check username, password and mobile number

9.2.1.4 Example Request

```
[POST] https://customer.identity.tm/api/2.10/requestNewPassword

{
    "mobile": "+49171123456"
}
```

9.2.2 Confirm password change

By executing this method, a new password will be set. The new password will be part of the response. If you provide the wrong TAN number, no new password will be set, and a password change needs to be requested again.

9.2.2.1 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/confirmNewPassword
```

9.2.2.2 JSON Request Body

Element	Parent	Type	Len	Req	Description
tan	[ROOT]	S	6	Y	Confirmation TAN

9.2.2.3 Response Data

Element	Parent	Type	Len	Occ	Description
password	[ROOT]	S		Y	The new password

9.2.2.4 HTTP Codes

200 OK, password has been changed, new password in body
400 Bad request, error description in body
401 Authentication failed, please check username, password, and TAN

9.2.2.5 Example Request

```
[POST] https://customer.identity.tm/api/2.10/confirmNewPassword

{
    "tan": "123456"
}
```

9.2.2.6 Example Response

```
{
    "password": "XXXXXXXXXXX"
}
```

9.3 Deep Linking into apps

The native **identity.TM** video/eID apps support deep linking.

If your users open such a link on a mobile device they will be redirected to the according app store (Google Play on Android devices and App Store on iOS devices). In case they open such a link on an unsupported mobile or a desktop device the usual **identity.TM** landing page will

be shown. The prime advantage for the user is that he or she no longer needs to find the right app manually and enter his or her credentials into it.

To receive such a deep link, which is a simple URL after all, you need to set the flag `getDeepLink` to true when executing the `putOrder` request. It is up to you to either redirect the user directly using the received deep link URL, putting it into an e-mail or presenting the link to the user as a QR code.

9.4 Optional content encryption

Several API calls that deal with verified identification data or recorded data support an optional content encryption scheme for additional security. To get the data in an encrypted form, you only need to append `/crypt` to the URL.

Keep in mind, that you need to share an AES-256 encryption key with us to use this function. This key will be used to encrypt the data.

Trying to call an API function with the `/crypt` argument without having your account properly set up will result in an error code.

The data format will be:

- Initialization vector, 23 bytes long
- Encrypted data
- GCM Tag

The content type header will remain unaltered.

If necessary, we can provide you with a code snippet, showing how to decrypt the encrypted data.

9.5 Testrobot

To activate this feature, set a specific value to the **Autotest** or to the **Ref** (*deprecated*) element during order creation. However, if **Autotest** is set, the **Ref** value will not affect this feature.

The following values are supported for the **Autotest** element:

```
POSITIVE
NEGATIVE
FRAUD
SIGN
```

In case when you use the deprecated way to activate this feature via **Ref** element then the following values are supported:

```
X-AUTOTEST-POSITIVE
X-AUTOTEST-NEGATIVE
```

X-AUTOTEST-FRAUD
X-AUTOTEST-SIGN

**The difference between the values of the two elements has no functional impact, and the outcome is the same.*

As a result, a respective **testrobot** flow will be triggered. The order will be automatically marked as completed approximately 20 seconds after creation, and the corresponding status codes based on the value set in the **Autotest(**or **Ref**) element will be assigned to the order. The provided **IdentData** upon order creation will be available for retrieval via the **getIdentData** endpoint once the order is completed (after 20 seconds). The returned result will match the previously provided **IdentData**, with the single exception that the ID card images are set to our sample images, which are also included in the **getIdentData** response.

For eSignPos orders (Add 4194304), and only for them, you can use SIGN as flow.

For eID test cases, the eidRestrictedIdentifier is generated randomly unless you specify IdentData->IDCard->CardNo. If this is the case, it is generated on the basis of the CardNo specified.

This feature is available for integration accounts only.

Example of order creation:

Request

```
[POST] https://customer.identity.tm/api/2.10/putOrder

{
        "Ref": "Test order",
        "Autotest": "POSITIVE",
        "Product": 19,
        "IdentData": {
            "Email": "mustermann@idnow.io",
            "Gender": "male",
            "Firstname": "Hans",
            "Surname": "Mustermann",
            "Street": "Musterstr 10",
            "ZipCode": "40231",
            "City": "Düssledorf",
            "Country": "DE",
            "Birthdate": "1980-01-01",
            "Birthplace": "Düssledorf",
            "Nationality": "DE",
            "CollectedBy": "Mustermann",
            "IDCard": {
                "CardNo": "f8b2931c-46b0-477f-bad5-23649a9bb948",
                "PlaceOfIssue": "Düssledorf",
                "DateOfIssue": "2010-12-12",
                "ValidUntil": "2030-12-12",
                "Type": "IDCard",
                "CountryOfIssue": "DE"
            }
        }
}
```

Response


```
{
    "OrderID": "9992403897831D",
    "Ref": "Test order"
}
```


Example of retrieving IdentData:

Request

```
[GET] https://customer.identity.tm/api/2.10/getIdentData/9992403897831D
```

Response


```
{
    "Identitfied": "2025-05-15T15:25:31+02:00",
    "OcrStatus": 0,
    "eIdUsed": false,
    "SecurityCheckFailed": false,
    "IdentData":{
        "0": {
            "Type": 1,
            "Firstname": "Hans",
            "Surname": "Mustermann",
            "Street": "Musterstr 10",
            "ZipCode": "40231",
            "City": "Düssledorf",
            "Country": "DE",
            "Birthdate": "1980-01-01",
            "Birthplace": "Düssledorf",
            "Birthname": null,
            "Nationality": "DE",
            "Gender": "male",
            "Email": "mustermann@idnow.io",
            "IDCard": {
                "CardNo": "f8b2931c-46b0-477f-bad5-23649a9bb948",
                "PlaceOfIssue": "Düssledorf",
                "DateOfIssue": "2010-12-12",
                "ValidUntil": "2030-12-12",
                "Type": "IDCard",
                "CountryOfIssue": "DE"
            }
        },
        "Images": [
            {
                "Image": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wC...",
                "Filename": "front_20250516-085942-865344415.jpg"
            },
            {
                "Image": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wC...",
                "Filename": "back_20250516-085942-558834056.jpg"
            },
            {
                "Image": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wC...",
                "Filename": "face_20250516-085942-657504955.jpg"
            }

        ],

    }

}
```

10 Appendix

10.1 IP-Ranges currently used

Range	Network name
212.211.189.136 - 212.211.189.143	ID8_NET
212.211.206.240 - 212.211.206.247	identity-NET
212.211.181.64 - 212.211.181.71	idvos_NET
212.211.191.96 - 212.211.191.103	ID8_NET

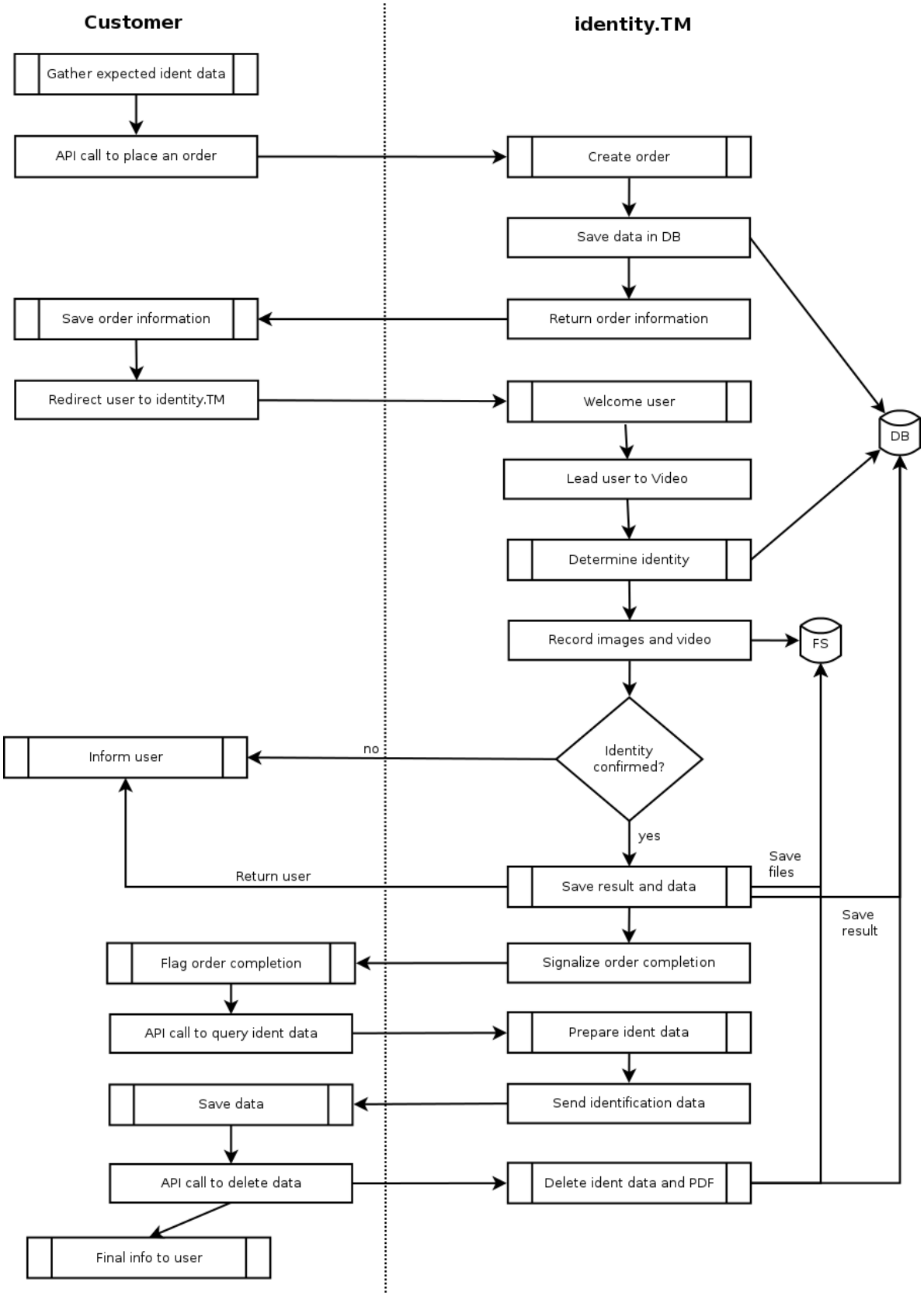
10.2 Supported browsers

- **Desktop**
  - Windows: Edge, Chrome, Opera, Firefox – latest two versions
  - macOS: Chrome, Opera, Firefox, Safari – latest two versions
- **Mobile**
  - Android: Chrome – latest official version
  - iOS: Safari – latest official version

10.3 Technical requirements for Video Ident

- A supported browser, alternatively mobile apps for Android and iOS can be used
- Bandwidth: Minimum 0,5 MB/s up/down
- Camera: Minimum resolution 640 x 480 px
- Network requirements:
  - Minimum: The minimum requirement is that TCP port 443 is open. Some firewall/proxy rules only allow for SSL traffic over port 443. You will need to make sure that non-web traffic can also pass over this port. TLS1.2
  - Better Experience: In addition to the minimum requirements being met, we also recommend that UDP port 3478 is open. TLS1.2
  - Best Experience: For the best experience possible, we recommend that UDP ports 1025 - 65535 be open. TLS1.2
- WebRTC: Outbound TCP, non-SSL web traffic on port 443 and the following domains must be accessible:
  - *.tokbox.com
  - static.opentok.com
  - enterprise.opentok.com
  - api.opentok.com
  - anvil.opentok.com
- WebSocket: In some situations, WebSocket connections are blocked over port 80. In this case a secure SSL connection using WSS over port 443 should successfully connect. The destinations and ports used by Pusher clients are as follows:
  - *.pusherapp.com on ports 80 and 443
  - *.pusher.com on ports 80 and 443

10.4 Simplified flow of a video identification



10.5 Simplified Video + eSign flow